CLAIMS

1. A software rewriting method that detects unexecuted parts of software to be rewritten during execution of the software to be rewritten and rewrites the unexecuted parts sequentially.

2. The software rewriting method according to claim 1, wherein the software to be rewritten is software having one block or software, which is divided into a plurality of blocks, and the respective blocks are classified into executing blocks and unexecuted blocks, and the unexecuted blocks are sequentially rewritten.

3. The software rewriting method according to claim 2, wherein rewriting blocks are temporarily stored in a memory, the rewriting blocks are compared with the executing blocks, and when blocks corresponding to the rewriting blocks are unexecuted, the corresponding blocks of the software to be rewritten are sequentially rewritten to the rewriting blocks.

4. The software rewriting method according to claim 3, wherein it is determined as to whether or not the corresponding blocks of the software to be rewritten are finished rewriting, and no rewriting of the corresponding blocks, which have been rewritten, are carried out again.

5.A software rewriting apparatus comprising:

a software storage for storing software having one block or a plurality of divided blocks;

a processor for expanding the blocks to be executed;

a block storage for temporarily storing rewriting blocks;

a discriminator for comparing the rewriting blocks with blocks executed by the processor to discriminate execution states of the blocks corresponding to the rewriting blocks; and

a rewriter for performing rewrite processing in which the corresponding blocks stored in the software storage are sequentially rewritten to the rewriting blocks in accordance with a discrimination result.

6. The software rewriting apparatus according to claim 5, wherein the discriminator comprises a table including configuration items having an item indicative of block numbers of the rewriting blocks and an item indicative of execution states of the blocks corresponding to the rewriting blocks, and the rewriter performs rewrite processing with reference to the table.

7. The software rewriting apparatus according to claim 6, wherein the table including a configuration item having an item indicative of rewriting states of the blocks corresponding to the rewriting blocks.

8. The software rewriting apparatus according to claim 5, further comprising a controller for surveying a degree of load of processing executed

by a CPU, and instructing the rewriter to carry out rewrite processing when the degree of load becomes low.

9. The software rewriting apparatus according to claim 8, wherein the controller surveys the degree of load of processing executed by the CPU in response to a rewrite request sent from the rewriter.

10. A communication terminal apparatus having a software rewriting apparatus, said software rewriting apparatus comprising:

a software storage for storing software having one block or a plurality of divided blocks;

a processor for expanding the blocks to be executed;

a block storage for temporarily storing rewriting blocks;

a discriminator for comparing the rewriting blocks with blocks executed by the processor to discriminate execution states of the blocks corresponding to the rewriting blocks; and

a rewriter for performing rewrite processing in which the corresponding blocks stored in the software storage are sequentially rewritten to the rewriting blocks in accordance with a discrimination result.